



Digitaalisen median tekniikat

Luento 5: AJAX & JavaScript-kirjastot

Luennot

1. Intro
2. XHTML
3. CSS
4. JavaScript
5. **AJAX & JavaScript-kirjastot**
 - **AJAX**
 - **JSON**
 - **Kokeillaan AJAX + JSON**
 - **Kirjastojen esittely**
 - **Kokeillaan kirjastoa**
6. Käytettävyys & saavutettavuus
7. Palvelinohjelmointi (PHP)
8. PHP jatkuu + pelkkiä kysymyksiä luento
9. Video- ja kuvaformaatit
10. Quirks
11. Hyviä, mutta vähän käytettyjä web-standardeja
12. Yhteenveto

Miten tunnistetaan JS tuki?

```
function jsEnabled() {  
    return true;  
}
```

```
if (jsEnabled()) {  
    // tee asioita  
}
```

AJAX

- Asynchronous JavaScript and XML
 - Asynkroninen tiedonsiirto ”selaimen sisällä”
- Ei itsenäinen tekniikka, vaan kokoelma käytäntöjä
- Voidaan siirtää muutakin dataa, kuin XML:ää
 - XHTML, JSON, teksti...
- Nopeuttaa sivun lataamista
- Siirtymät eivät näy selaimen historiassa
- Menetetään osoitettavuus!
 - Ajaxia on käytettävä tarkoin.

XMLHttpRequest -olio

- Tästä lähtien: XHR
- Olion avulla sisällön haku on mahdollista tehdä asynkronisesti: ei jäädä odottamaan vastausta
- Kun olion tila muuttuu, saadaan tieto ja voidaan suorittaa JavaScriptiä
- Esimerkki [reaaliaikainen haku](#)

JSON

- JavaScript Object Notation
- JavaScript –olio tekstimuotoisena esityksenä
- Tuodaan ”olioavaruuteen” käyttämällä JavaScriptin (**vaarallista**) `eval()` –funktiota.
 - Tämä ongelma voidaan korjata itse
 - ```
var my_JSON_object = !(/^;{}\[0-9.\-+Eaeflnr-u \n\r\t]/.test(text.replace(/\"([^\"])*\"/g, ""))) && eval('(' + text + ')');
```
  - TAI sitten vaan käytetään kirjastoja (kohta lisää)

# JSON esimerkki

Yksittäinen: olio["name"]

```
{ "name": "Waterloo",
 "artist": "Abba",
 "duration": 193 }
```

Taulukko: olio["songs"][indeksi]

```
{ "songs": [
 { "name": "Waterloo",
 "artist": "Abba",
 "duration": 193 },

 { "name": "Dancing Queen",
 "artist": "Abba",
 "duration": 242 }] }
```

# JSON vs XML

```
{ "menu": {
 "id": "file",
 "value": "File",
 "popup": {
 "menuitem": [
 { "value": "New", "onclick": "CreateNewDoc()" },
 { "value": "Open", "onclick": "OpenDoc()" },
 { "value": "Close", "onclick": "CloseDoc()" }
]
 }
}
```

Sama XML:nä

```
<menu id="file" value="File">
 <popup>
 <menuitem value="New" onclick="CreateNewDoc()" />
 <menuitem value="Open" onclick="OpenDoc()" />
 <menuitem value="Close" onclick="CloseDoc()" />
 </popup>
</menu>
```

# Kokeillaan käytännössä

- Ajax esimerkki
- JSON esimerkki

# JavaScript-kehyykset

- Helpottavat elämää, pyörää ei tarvitse keksiä uudestaan
- Selainyhteensopivuus ja suorituskyky on jo mietitty (ja testattu)
- Pelkät kirjastot eivät kuitenkaan riitä. On tärkeää ymmärtää, miten AJAX ja JavaScript toimivat.
- Kirjastoja on mm:
  - jQuery
  - Prototype
  - Dojo
- Luennolla esitellään jQuery, muissakaan ei varsinaisesti ole mitään vikaa

# Perusteita

- Kirjasto ladataan käyttöön `<script>`:llä
- `document.getElementById("container");`
  - On jQueryllä: `$("#container");`
- `$("#container").hide();`
- `$(".luokka");`
  - Taulukko elementeistä, joilla on luokka
- `$("#container > .luokka");`
  - "container" id:n lapset joilla on class "luokka"

# Ketjutus

- Rankka ketjutus mahdollista. Tätä kannattaa sulatella hetki ja kokeilla itse:

```
$('#taulu').click(function(event) {
 $('#kuvateksti').hide();
 $(this).fadeOut(function() {
 $('ul').fadeOut();
 });
 event.preventDefault(); // EI return(false); !
});
```

# jQuery webissä

- <http://www.jquery.com>
  - Hyviä tutoriaaleja
- <http://api.jquery.com>
  - Kätevä API
- Nyt katsotaan esimerkkejä (tulevat kurssin kotisivuille myös)